General QoS-Aware Scheduling Procedure for Passive Optical Networks

A General QoS-Aware Scheduling Procedure for Passive Optical Networks

Mohammad Hadi, Chayan Bhar, and Erik Agrell, *Fellow, IEEE*

***Abstract*—Increasing volume, dynamism, and diversity of the access traffic have complicated the challenging problem of dynamic resource allocation in passive optical networks. We introduce a general scheduling procedure for passive optical networks, which optimizes a desired performance metric for an arbitrary set of operational constraints. The proposed scheduling has a fast and causal iterative implementation, where each iteration involves a local optimization problem followed by a recursive update of some status information. The generality of the platform enables a proper description of the diverse quality of service requirements, while its low computational complexity makes agile tracking of the network dynamism possible. To demonstrate its versatility and generality, the applications of the scheme for service-differentiated dynamic bandwidth allocation in time- and wavelength-division-multiplexed passive optical networks are discussed. To further reduce the computational complexity, a closed-form solution of the involved optimization in each iteration of the scheduling is derived. We directly incorporate transmission delay in the scheduling and show how the consumed power is traded for the tolerable amount of transmission delay. Furthermore, a 50% power efficiency improvement is reported by exploiting the inherent service diversity among the subscribers. The impact of service prioritization, finite buffer length, and packet drops on the power efficiency of the scheme is also investigated.**

***Keywords*—dynamic scheduling, passive optical networks, power efficiency, quality of service**

I. Introduction

PASSIVE optical networks (PONs) play an essential role in developing power-efficient and quality of service (QoS)-aware access networks [1, sec. 2], [2], [3]. Video services and 5G applications have been considered as the strongest drivers for changing traffic paradigms in PONs. In fact, video delivery over PON deals with high capacity demands and diverse QoS requirements [4]. On the other hand, PON is one of the contenting technologies for 5G transport, and therefore, the dynamic and diverse behavior of the 5G traffic can have a significant impression on traffic paradigms in PON [5], [6]. These changing traffic patterns have imposed new challenges on the complex problem of dynamic resource allocation in PONs [1, sec. 5] and consequently, motivated development of new network control schemes [2], [4]. Unfortunately, network management in current PONs is not yet mature to address the increasing volume, dynamism, and diversity of the future, and even today's, traffic streams. In fact, a proper scheduling in PONs needs a fine and agile adaptation between the allocated resources and dynamic network behavior. This adaptation is more important for a heterogeneous PON with different service requirements among its subscribers. Through the separation of control and data plane, software-defined networking (SDN) provides a dynamically fine-grained traffic control that enhances total network controllability and manageability [7], [8]. In an SDN-based PON, the data plane can be any type of available PONs such as gigabit PON (G-PON), time-division-multiplexed PON (TDM-PON), time- and wavelength-division-multiplexed PON (TWDM-PON), and code-division multiple-access PON (CDMA-PON) [1, sec. 2], [7]–[9]. A unified SDN control plane needs a general scheduling procedure to handle resource management for different implementations of the data plane. The importance of this general scheduling scheme is doubled by the growing tendency to develop more diverse, tunable, and high-speed devices for next-generation PONs (NG-PONs). Although there are many customized scheduling methods for different PONs [2], [9]–[12], to the best of our knowledge, there is no research work on a general scheduling platform to allocate resources of a PON, regardless of its implementation technology.

In [13], we introduced Lyapunov optimization for iterative scheduling in a CDMA-PON system, where transceivers are periodicity reconfigured such that the transmission rate is maximized while queue stability and bit error rate constraints are satisfied. The strategy was generalized and applied to elastic optical networks in [14]. In this work, we use Lyapunov optimization to develop a general scheme to iteratively schedule a PON, regardless of its implementation technology. With Lyapunov optimization, a differential measure of the PON status and its performance metrics describes a structured objective function. Each iteration of the Lyapunov method involves an optimization of the structured objective subject to a given set of operational constraints [15], [16]. Unlike pure heuristic scheduling algorithms [17]–[20], the proposed iterative solution is supported by mathematical analysis. Furthermore, its general structure allows to describe not only the instantaneous but also the time-averaged behavior of the PON by equality and inequality constraints [18], [21]. Moreover, important concepts such as service prioritization, traffic classification, traffic shaping, and queueing stability can be effectively described and handled in the proposed scheduling platform [15], [16].

Application examples for TDM- and TWDM-PONs are provided in this paper. Especially, we demonstrate the application of the proposed method for power-efficient service-differentiated scheduling in a TDM-PON, and then extend the results to support a TWDM-PON. As an important contribution, we derive a closed-form solution of the involved

The authors are with Department of Electrical Engineering, Chalmers University of Technology, Sweden, e-mail: mohadi@chalmers.se. This work was supported in part by Vinnova under grant no. 2017-05228 and the Knut and Alice Wallenberg Foundation under grant no. 2013.0021.

optimization problem in each iteration of the scheduling. This closed-form solution considerably expedites the scheduling and provides much adaptation capability. The average transmission delay, as an important QoS parameter, is an explicit input to the closed-form solution, which can be set to any value for facilitating the desired QoS constraint. The scheduling tries to consume the minimum possible power to meet transmission delay requirements at every offered load. The smoothness of the simulated transmission delay curves confirms the capability of the scheme for guaranteeing a desired value of the transmission delay. Furthermore, simulation results show that the scheduling method assigns the available resources according to the required QoS levels and priorities, and exploits the existing service diversity among the subscribers to improve the power efficiency. The scheduling also offers adjustable tradeoffs between power efficiency, transmission delay, and drop rate to reduce the consumed power at the cost of a controllable increase in average transmission delay or drop rate. To provide practical results, the simulated PON is loaded by a realistic self-similar traffic pattern, and the impact of physical limitations, such as finite buffer storage and packet drops, on the performance metrics is investigated.

The paper evolves by introducing the general scheduling procedure in Sec. II. The application of the proposed scheme is demonstrated in Sec. III, where a power-efficient service-differentiated scheduling is proposed. Simulation results are provided in Sec. IV. Finally, concluding remarks are delineated in Sec. V.

II. General Scheduling

With a few minor terminology changes, the general resource allocation scheme proposed in [14] can be used for scheduling in PONs, as a novel customized application with its own operational constraints and computational requirements. However, to make the paper standalone, a brief description is provided. Consider a general PON, which operates in discrete time intervals $n \in \mathbb{Z}_0^\infty$, where $\mathbb{Z}_a^b$ includes the integers between a and b inclusively. There are I entities (such as ONUs), each having a queue with backlog $q_i[n]$, which is filled and emptied by $a_i[n]$ arrived and $s_i[n]$ departed bits in every interval n. The served bits $s_i[n]$ are determined by reconfiguring the resource vector $\boldsymbol{\alpha}[n]$ (with possible elements such as transmission window) in each interval n such that a targeted performance metric (such as power consumption) is optimized, while some QoS requirements (such as average transmission delay) are guaranteed and operational constraints (such as guard time) are satisfied.

In each interval n, the resource vector $\boldsymbol{\alpha}[n]$ is reconfigured by solving the optimization problem

$$\min_{\boldsymbol{\alpha}[n]} \quad \Gamma f_0[n] + \textstyle\sum_{i\in\mathbb{Z}_1^I} q_i[n](a_i[n]-s_i[n]) + \sum_{l\in\mathbb{Z}_1^{L_1}} p_{l,1}[n] f_{l,1}[n] + \sum_{l\in\mathbb{Z}_1^{L_2}} p_{l,2}[n] f_{l,2}[n] \quad \text{s.t.} \quad (1a)$$

$$f_{l,3}[n] \leqslant 0, \quad \forall l \in \mathbb{Z}_1^{L_3}, \quad (1b)$$

$$f_{l,4}[n] = 0, \quad \forall l \in \mathbb{Z}_1^{L_4}, \quad (1c)$$

where $f_0[n]$, $f_{l,m}[n]$, and $s_i[n]$ are arbitrary functions of $\boldsymbol{\alpha}[n]$ and the Lyapunov penalty coefficient Γ controls the balance between $f_0[n]$ and the summation terms in (1a). The actual queue length $q_i[n]$ and the so-called virtual queue length $p_{l,m}[n]$ are updated recursively as

$$q_i[n+1] = \max\{q_i[n] + a_i[n] - s_i[n], 0\}, \quad i \in \mathbb{Z}_1^I, \quad (2a)$$

$$p_{l,1}[n+1] = \max\{p_{l,1}[n] + f_{l,1}[n], 0\}, \quad l \in \mathbb{Z}_1^{L_1}, \quad (2b)$$

$$p_{l,2}[n+1] = p_{l,2}[n] + f_{l,2}[n], \quad l \in \mathbb{Z}_1^{L_2}, \quad (2c)$$

with initialization $q_i[0] = 0$, $i \in \mathbb{Z}_1^I$ and $p_{l,m}[0] = 0$, $m \in \mathbb{Z}_1^2$ and $l \in \mathbb{Z}_1^{L_m}$. Optimization (1) approaches the minimized time-averaged objective $\overline{f_0}$ with a proximity controlled by Γ, where the overbar denotes time average defined as $\overline{x} = \lim_{n\to\infty} \frac{1}{n} \sum_{n'\in\mathbb{Z}_0^{n-1}} \mathcal{E}\{x[n']\}$ with an expectation over random arrival events. As proven in [16] using the Lyapunov analysis, when the arrival process $a_i[n]$ is independent and identically distributed (i.i.d.) over intervals and $q_i[n]$ does not appear in the functions $f_{l,m}[n]$, then the first penalty term in (1a) and update equation (2a) assure the queue stability by imposing $\overline{a_i} \leqslant \overline{s_i}$ to keep queue lengths finite over time. Similarly, the second and third penalty terms along with their corresponding update equations (2b) and (2c) enforce $\overline{f_{l,1}} \leqslant 0$ and $\overline{f_{l,2}} = 0$, respectively, which can be used to describe various QoS and operational constraints using time-averaged inequality or equality expressions. Moreover, (1b) and (1c) define additional instantaneous inequality and equality constraints to be satisfied in each interval.

When $a_i[n]$ is not i.i.d. or $q_i[n]$ appears in $f_{l,m}[n]$, the performance of (1) is not rigorously guaranteed. However, as demonstrated numerically, the proposed general scheme yields acceptable results for many practical scenarios with even more relaxed assumptions [14]. To support this claim and show the applicability of the scheme for PONs, we provide in the next section a few application examples and use simulation results to validate their performance.

III. Application Examples

The involved intuition in the general scheme inspires the development of novel iterative and practical scheduling procedures, such as the proposed scheme in [13] for CDMA-PONs. To further affirm this assertion, a power-efficient service-differentiated scheduling scheme for TDM-PONs is established, and then extended to support TWDM-PONs. The performance of the scheme is mathematically guaranteed in some special cases [15], [16], as briefly explained in Sec. II, while its general performance can be evaluated using simulation results, as will be discussed in Sec. IV. The main notation is summarized in Table I. The time index n of the variables is removed in coming equations and Table I to emphasize that the scheme only needs to store the status of the current interval.

A. TDM-PON

Consider the TDM-PON of Fig. 1(a) with a gross bit rate of R_U in the upstream direction. The time is divided into intervals with duration T_C. The upstream capacity of the PON in an interval is time-shared by I ONUs. An ONU can operate in active or sleep mode, with or without transceiving capability,

TABLE I: Main constants and variables.

Type	Notation	Definition
Constants	$I = 32$	Number of ONUs
	$V_i = 100$	Drop penalty
	$N_W = 1$	Number of wavelengths
	$\Gamma = 10$	Lyapunov penalty
	$R_U = 10$ [Gbit/s]	Upload bit rate
	$T_C = 2$ [ms]	Interval duration
	$T_D = 0$ [ms]	Maximum round trip time difference
	$T_H = 51.2$ [ns]	Report time
	$T_G = 1$ [us]	Guard time
	$T_O = 2$ [ms]	Transition time
	$T_S = 0$ [us]	Start time
	$T_P = 0$ [us]	Process time
	$T_W = 50$ [us]	Wavelength tuning time
	$P_S = 0.75$ [W]	Sleep power
	$P_A = 4.2$ [W]	Active power
	$T_i = 80$ [us]	Round trip time
	$D_i = 6$ [ms]	Average delay
	$Q_i = 8$ [Mbit]	Delaying buffer capacity
	$E_i = 1$ [Mbit]	Maximum arrived bits in an interval
	$A_i = 1.5$ [Mbit]	Collecting/Shaping buffer capacity
Variables	$\mathbb{A}_w \subset \mathbb{A}$	Active ONU set per wavelength
	e_i	Wavelength number
	p_i	Virtual queue backlog
	q_i [bit]	Delaying buffer backlog
	a_i [bit]	Shaping buffer backlog
	s_i [bit]	Served bits
	b_i [bit]	Uploaded bits
	d_i [bit]	Dropped bits
	c_i [bit]	Number of sleep intervals
	t_i [s]	*GATE* transmission time

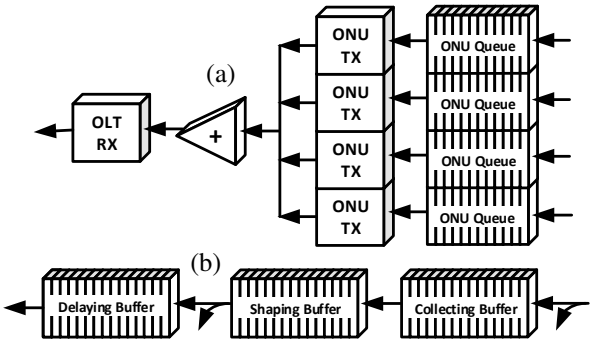

Fig. 1: (a) Upstream data flow for the TDM-PON application example. (b) The proposed ONU queue structure for the TDM-PON application example.

and different consumed power P_A or P_S, respectively. Each ONU transmitter i is fed by a limited-size queue. As illustrated in Fig. 1(b), the queue is in turn divided into three consecutive parts, named delaying buffer with storage capacity Q_i and backlog length q_i, shaping buffer with storage capacity A_i and backlog length a_i, and collecting buffer with a same storage capacity A_i as the shaping buffer. The small storage capacity A_i can accommodate the maximum number of bits arrived to ONU i during an interval E_i. A central scheduler residing in the OLT commands the ONUs using *GATE* messages while the ONUs narrate their status by sending *REPORT* messages to the OLT. The OLT sends a *GATE* message (b_i, d_i, c_i) to ONU i at time t_i and announces the number of uploaded bits from the delaying buffer b_i, the number of dropped bits from the shaping buffer d_i, and the number of sleep intervals c_i. The ONU i includes the backlog length of the delaying and shaping buffers, q_i and a_i, in its sent *REPORT* message (a_i, q_i) to the OLT.

The scheduler always remembers the latest *REPORT* messages (a_i, q_i) and stores the index of the active ONUs of the current interval in a set $\mathbb{A} \subset \mathbb{Z}_1^I$. As shown in Fig. 2, an interval begins when the OLT runs its scheduling algorithm. The scheduling, which needs the time T_P to be finished, includes three main steps of optimization (3), updating (5), and prediction (6), gets the latest *REPORT* messages (a_i, q_i) as the input, and determines the corresponding *GATE* messages (b_i, d_i, c_i) and proper transmission times t_i as the output. The ONUs require different average transmission delay D_i and the OLT involves these distinct QoS requirements in the scheduling. When the scheduling ends, the OLT sends each *GATE* message at its proper time t_i. After a propagation delay $T_i/2$, where T_i is the round trip time (RTT) of ONU i, the *GATE* message (b_i, d_i, c_i) reaches the ONU i. After the time T_S from receiving the *GATE* message, ONU i uploads the maximum number of the packets with a sum length no more than b_i bits from the delaying buffer to the OLT, drops the minimum number of packets with a sum length no less than d_i bits from the shaping buffer and moves the remaining bits to the delaying buffer, clears the collecting buffer for upcoming packets by moving its content to the shaping buffer, announces the new volumes of the delaying and shaping buffers to the OLT using the *REPORT* message (a_i, q_i) in time T_H, and then goes to sleep mode for c_i intervals. Since a transition from sleep to active mode needs the time T_O, each ONU starts its reactivation at a proper time before the end of its sleep interval, when the OLT will send a new *GATE* message. The OLT also considers a guard time T_G between consecutive uploads of the different ONUs.

Considering the required average delay D_i, the delaying buffer can postpone packet transmission to put the ONU in sleep mode and save power. Although there is no forced assumption on the delaying buffer storage capacity Q_i, intuitively, it should have a storage capacity proportional to the required delay D_i to achieve the expected power efficiency. The controllable amount of packet drops in the shaping buffer helps to satisfy very strict delay requirements and enables a useful trade-off between drop rate and power consumption. The storage capacity of the collecting buffer A_i should be enough to accommodate not only the maximum arrived bits in an interval E_i but also the arriving bits during sleep mode. When there is no free space in the collecting buffer, parts of the incoming packets are dropped. We refer to the packet drops in the shaping buffer as the controllable drops while packet drops due to a full collecting buffer are called unwanted drops. Such unwanted drops can be prevented by a suitable choice of A_i and E_i. For instance, when all ONUs have a same arrival process and equally share the upstream capacity, a reasonable choice is $A_i \gtrsim E_i$, and $E_i \gtrsim R_U T_C / I$. As a result, assuming a suitable value for A_i, we only focus on the controllable packet drops in the shaping buffer during different steps of the scheduling process.

In the optimization step, the OLT calculates b_i and d_i, $i \in \mathbb{Z}_1^I$ by solving

$$\min_{\mathbf{b},\mathbf{d}} \sum_{i \in \mathbb{Z}_1^I} \left[\Gamma(b_i + V_i d_i) + p_i (q_i - \frac{D_i}{T_C}(a_i - d_i)) \right] \quad \text{s.t.} \quad \text{(3a)}$$

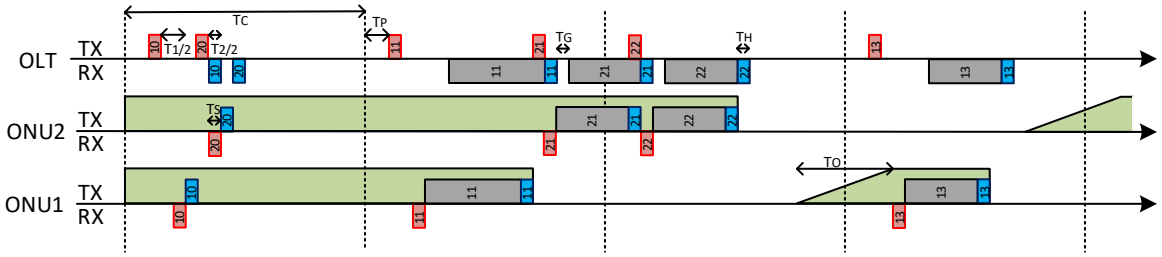

Fig. 2: Upstream timing diagram for a TDM-PON with two ONUs. *REPORT*, *GATE*, and data messages are shown by blue, red, and gray colors, respectively. Green color shows ONU active times. The numbers in each message show the ONU and interval indices. Different timing constants are also illustrated.

$$b_i \geqslant 0, d_i \geqslant 0, \quad i \in \mathbb{Z}_1^I, \tag{3b}$$

$$a_i + q_i - b_i - \min\left\{Q_i, \frac{D_i}{T_C} a_i\right\} \leqslant d_i, \quad i \in \mathbb{A}, \tag{3c}$$

$$\sum_{i \in \mathbb{A}} b_i \leqslant R_U \Big(T_C - T_D - |\mathbb{A}| \left(T_H + T_G\right)\Big), \tag{3d}$$

$$b_i = 0, d_i = 0, \quad i \in \mathbb{Z}_1^I \backslash \mathbb{A}, \tag{3e}$$

where $|\mathbb{A}|$ is the cardinality of $\mathbb{A}$. The optimization problem (3) is a simple adoption of (1) obtained by the substitutions $\boldsymbol{\alpha} = (\mathbf{b}, \mathbf{d})$, $L_1 = I$, $L_2 = 0$, $L_3 = 3I + 1$, $L_4 = 2I$,

$$s_i = b_i + d_i, \quad i \in \mathbb{Z}_1^I, \tag{4a}$$

$$f_0 = \sum_{i \in \mathbb{Z}_1^I} (b_i + V_i d_i), \tag{4b}$$

$$f_{i,1} = q_i - \frac{D_i}{T_C}\left(a_i - d_i\right), \quad i \in \mathbb{Z}_1^I, \tag{4c}$$

$$f_{i,3} = -b_i, f_{I+i,3} = -d_i, \quad i \in \mathbb{Z}_1^I, \tag{4d}$$

$$f_{2I+i,3} = 0, \quad i \in \mathbb{Z}_1^I \backslash \mathbb{A}, \tag{4e}$$

$$f_{2I+i,3} = a_i + q_i - s_i - \min\left\{Q_i, \frac{D_i}{T_C} a_i\right\}, \quad i \in \mathbb{A}, \tag{4f}$$

$$f_{3I+1,3} = \sum_{i \in \mathbb{A}} b_i - R_U \Big(T_C - T_D - |\mathbb{A}| \left(T_H + T_G\right)\Big), \tag{4g}$$

$$f_{i,4} = 0, f_{I+i,4} = 0, \quad i \in \mathbb{A}, \tag{4h}$$

$$f_{i,4} = b_i, f_{I+i,4} = d_i, \quad i \in \mathbb{Z}_1^I \backslash \mathbb{A}. \tag{4i}$$

Similar to (1a), the objective function (3a) has a main term multiplied by the Lyapunov coefficient Γ, and a penalty term to satisfy a time-averaged constraint. The main term $b_i + V_i d_i$ is a weighted sum of the upload bits b_i and dropped bits d_i. To shorten the upload time and equivalently, extend the sleep time and improve the power efficiency, the number of upload bits b_i is minimized in (3a). Furthermore, the priority of the packet delivery over packet drop can be adjusted by the coefficient V_i. Moreover, distinct V_i for different ONUs creates a service prioritization and allows to drop part of the stored packets of a low-priority ONU at high load conditions. The penalty term of (3a) ensures the time-averaged constraint $\overline{q_i} - D_i/T_C(\overline{a_i} - \overline{d_i}) \leqslant 0$, which is a simple adoption of Little's theorem to guarantee the required average delay D_i [22, sec. 1], [15]. In (3a), p_i is the length of the virtual queue corresponding to the averaged delay D_i. Constraint (3c) drops the incoming bits that cannot be accommodated in a storage size of $\min\left\{Q_i, D_i a_i / T_C\right\}$. Intuitively, the number of bits stored ahead of the new arrivals a_i should not exceed $D_i a_i / T_C$ to meet the required delay D_i. Therefore, we force an upper bound of $D_i a_i / T_C$ to the storage capacity of the delaying buffer to better control the transmission delay. Note that since the buffer length is finite in (3c), the queue stability is already guaranteed and there is no need to add the first penalty term of (1a) to (3a). Constraint (3d) specifies that the overall upload in an interval cannot exceed its net capacity. Note that the report and guard times are excluded in the calculation of the net capacity. Moreover, due to different distances from the OLT, an interval is seen with a variable time shift by each ONU. As can be interpreted from Fig. 2, to ensure that consecutive intervals do not overlap in the OLT and ONUs, the maximum RTT difference $T_D = \max_{i,i'} |T_i - T_{i'}|$ should also be excluded in constraint (3d). For inactive ONUs $i \notin \mathbb{A}$, $b_i = d_i = 0$ by constraint (3e). Fortunately, the OLT does not need to solve (3) numerically since its closed-form solution is available in (9), as proven in Appendix A. This closed-form solution considerably reduces the process time T_P and improves the scalability of the proposed scheduling, and should be consequently considered as an important novelty compared to other customized applications of the introduced general platform, such as [14].

In the updating step, (2b) is adapted to update the virtual queue length by

$$p_i \leftarrow \max\{0, p_i + q_i - \frac{D_i}{T_C}\left(a_i - d_i\right)\}, \quad i \in \mathbb{A} \tag{5a}$$

$$p_i \leftarrow \max\{0, p_i + q_i - \frac{D_i}{T_C} a_i\}, \quad i \in \mathbb{Z}_1^I \backslash \mathbb{A}, \tag{5b}$$

where a_i and q_i are set to their latest values derived from coming *REPORT* messages (a_i, q_i). Since the *REPORT* message automatically provides the latest value of q_i, the update equation (2a) is not needed.

In the prediction step, the number of intervals that ONU i can sleep is predicted by

$$c_i = \lfloor \max\left\{0, \min\left\{\frac{D_i}{T_C}, \frac{E_i}{a_i}\right\} - 1\right\} \rfloor, \quad i \in \mathbb{A}, \tag{6}$$

where $\lfloor . \rfloor$ is the floor function. In fact, (6) is intuitively obtained by considering the requirements on transmission delay and buffer overflow. To protect from overflow, the overall arrived bits to the collecting buffer during the sleep intervals should not exceed the capacity A_i. In the worst case, E_i bits arrive to ONU i in an interval and as assumed, the capacity A_i is

enough to accommodate these maximum arrived bits $E_i \lesssim A_i$. Now, if $a_i \leqslant E_i$ bits arrive in each interval, on average, it takes E_i/a_i intervals to have E_i bits in the collecting buffer. Therefore, the ONU can sleep for $E_i/a_i - 1$ intervals, and then, become active to serve the collected bits in the interval after the sleep without an overflow in the collecting buffer. On the other hand, it is not reasonable to have a sleep time more than the required average delay value, so $c_i T_C \leqslant D_i$. Note that the described intuition is based on the average behavior of the traffic and burst arrivals still may create unwanted drops in the collecting buffer. As stated, such unwanted drops are suppressed by proper selection of the constants A_i and E_i. Different ONUs can also have distinct E_i or D_i in (6) to let exploit the available diversity among the ONUs to improve network performance.

Assume that the upload priority within an interval is with the active ONU having the largest RTT. Let $\mathcal{Y}(h), h = 1, \ldots, |\mathbb{A}|$ return the index of the ONU having the hth largest RTT in the active set $\mathbb{A}$. As can be perceived from Fig. 2, the first *GATE* message is sent at $t_{\mathcal{Y}(1)} = nT_C + T_P$, where nT_C is the beginning time of interval n. Sequentially for $h = 2, \ldots, |\mathbb{A}|$, $t_{\mathcal{Y}(h)} = nT_C + T_P + T_{\mathcal{Y}(1)} - T_{\mathcal{Y}(h)} + \sum_{h'=1}^{h-1}(b_{\mathcal{Y}(h')}/R_U + T_G + T_H)$. Referring to Fig. 2, ONU i can sleep for $(n + c_i)T_C + T_P + T_i + T_G - T_O - t_i - b_i/R_U - T_H$, after which its reactivation takes time T_O. Note that we implicitly have $T_C - T_D - I(T_H + T_G) \geqslant 0$ to maintain the feasibility of the optimization problem in each interval. The special case of Algorithm 1 for $N_W = 1$ corresponds to the control scheme described above.

B. TWDM-PON

Upload transmission in a TDWM-PON with the upload bit rate R_U per each of N_W wavelengths can be similarly scheduled using the proposed general scheme. In fact, a TWDM-PON consists of N_W individual TDM-PON subsystems stacked into a common passive network using several wavelengths [23], [24]. Therefore, the described TDM-PON scheduling can be used for each wavelength. Let T_W denote the wavelength tuning time in tunable ONUs and add the new element e_i to the *GATE* message (b_i, d_i, c_i, e_i) to declare the operating wavelength of ONU i. Following the same way as in the previous subsection, the describing equations of the scheduling scheme can be derived, as summarized in Algorithm 1. Clearly, when $N_W = 1$, Algorithm 1 corresponds to the proposed TDM-PON scheduling scheme. After collecting the *REPORT* messages (a_i, q_i), in each iteration of Algorithm 1, a few local variables are initialized and the ONUs are sorted decreasingly by $x_i = V_i + p_i D_i/(T_C\Gamma)$ to compute $\mathcal{X}(h), h = 1, \ldots, |\mathbb{A}|$, which returns the index of the ONU having the hth largest coefficient x_i in the active set $\mathbb{A}$. Then, an ONU with $x_i > 1$ and $y_i = a_i + q_i - \min\{Q_i, D_i a_i/T_C\} > 0$ is assigned y_i bits from the remaining transmission capacity of the first unoccupied wavelength w. To reduce the number of used wavelengths and consequently the power consumption, the highest possible number of ONUs share the chosen wavelength w. If there are not enough resources to serve an ONU in wavelength w, the next wavelength $w + 1$ is selected. When *GATE* messages (b_i, d_i, c_i, e_i) are determined, the indices of active ONUs assigned a particular wavelength w are grouped in corresponding sets $\mathbb{A}_w$. Thereafter, each $\mathbb{A}_w$ is sorted by decreasing RTT. Finally, the transmission times of the *GATE* messages t_i are determined, sleep times c_i are calculated, and the status variables are updated. Upon receiving a *GATE* message, each ONU tunes its transmitter to the declared wavelength during the start time $T_S = T_W$ and follows the same buffering mechanism as the described TDM-PON buffering policy in Sec. III-A.

Algorithm 1: Scheduling Algorithm for TWDM-PON.

input: *REPORT* messages $(a_i, q_i), i \in \mathbb{Z}_1^I$
output: *GATE* messages $(b_i, d_i, c_i, e_i), i \in \mathbb{Z}_1^I$

for $n = 1, 2, \cdots$ **do**
- $b_i \leftarrow 0, d_i \leftarrow 0, i \in \mathbb{Z}_1^I$
- $\mathbb{A}_w \leftarrow \varnothing, w = 1, \cdots, N_W$
- $w \leftarrow 1$
- $z \leftarrow R_U\big(T_C - T_D - |\mathbb{A}|\,(T_H + T_G)\big)$
- $\mathbb{A} = \{i \in \mathbb{Z}_1^I | c_i = 0\}$
- $x_i \leftarrow V_i + p_i D_i/(T_C\Gamma), i \in \mathbb{A}$
- $y_i \leftarrow a_i + q_i - \min\{Q_i, D_i a_i/T_C\}, i \in \mathbb{A}$
- sort $\mathbb{A}$ decreasingly by x_i to compute $\mathcal{X}(.)$
- **for** $i = 1, \cdots, |\mathbb{A}|$ **do**
  - $b_{\mathcal{X}(i)} \leftarrow 0$
  - **if** $y_{\mathcal{X}(i)} > 0$ **and** $x_{\mathcal{X}(i)} > 1$ **then**
    - $b_{\mathcal{X}(i)} \leftarrow y_{\mathcal{X}(i)}$
  - **end**
  - **if** $b_{\mathcal{X}(i)} > z$ **and** $w < N_W$ **then**
    - $w \leftarrow w + 1$
    - $z \leftarrow R_U\big(T_C - T_D - (|\mathbb{A}| - i + 1)(T_H + T_G)\big)$
  - **end**
  - $b_{\mathcal{X}(i)} \leftarrow \min\{b_{\mathcal{X}(i)}, z\}$
  - $e_{\mathcal{X}(i)} \leftarrow w$
  - $\mathbb{A}_w \leftarrow \mathbb{A}_w \cup \{\mathcal{X}(i)\}$
  - $z \leftarrow z - b_{\mathcal{X}(i)}$
- **end**
- **for** $w = 1, \cdots, N_W$ **do**
  - sort $\mathbb{A}_w$ decreasingly by RTT to compute $\mathcal{Y}(.)$
  - $t \leftarrow 0$
  - **for** $i = 1, \cdots, |\mathbb{A}_w|$ **do**
    - $t_{\mathcal{Y}(i)} \leftarrow nT_C + T_P + T_{\mathcal{Y}(1)} - T_{\mathcal{Y}(i)} - t$
    - $t \leftarrow t + b_{\mathcal{Y}(i)}/R_U + T_G + T_H$
  - **end**
- **end**
- $d_i \leftarrow \max\{0, y_i - b_i\}, i \in \mathbb{A}$
- $p_i \leftarrow \max\{0, p_i + q_i - D_i\big(a_i - d_i\big)/T_C\}, \quad i \in \mathbb{A}$
- $p_i \leftarrow \max\{0, p_i + q_i - D_i a_i/T_C\}, \quad i \in \mathbb{Z}_1^I \backslash \mathbb{A}$
- $c_i \leftarrow \lfloor \max\{0, \min\big\{D_i/T_C, E_i/a_i\big\} - 1\big\} \rfloor, \quad i \in \mathbb{A}$
- $c_i \leftarrow \max\{c_i - 1, 0\}, \quad i \in \mathbb{Z}_1^I$

end

IV. Numerical Results

Transmission delay, power efficiency, and drop rate are the key performance metrics for evaluating a power-efficient PON scheduling scheme. Here, we report these performance metrics in terms of offered load for different values of a constant parameter to analyze the operation of the proposed scheme in different scenarios. Transmission delay is the average time

between a packet arrival to an ONU and its delivery to the OLT, and includes queuing delays in the different buffers as well as propagation delay. The power efficiency is the amount of saved power compared to a scenario where all the ONUs are permanently active. Drop rate measures the percentage of the packets dropped from the overall arrived packets to the ONUs. The offered load $\Omega = \sum_{i \in \mathbb{Z}_1^I} \overline{\omega_i}/R_U$ is the summation of the average bit-rate $\overline{\omega_i}$ injected to the PON by each ONU i divided by the upstream rate R_U. Unless otherwise mentioned, the constant parameters are set to their default values in Table I [10], [23], [25], [26]. The input traffic stream to each ONU is modeled from a self-similar traffic pattern [26]. Traffic demands of random sizes arrive at random times. The number of packets in each such demand is Pareto-distributed with shape parameter 1.25 and scale parameter 1, whereas the silence duration between two consecutive demand arrivals lasts for the time needed to send a virtual demand, whose number of packets is Pareto-distributed with shape parameter 1.25 and the scale parameter set to provide the desired value of Ω. The length of each packet is randomly and independently selected from a uniform distribution over $[512, 12144]$ bits [27].

For $N_W = 1$, the introduced method is a G-PON scheduling and comparable with the "Fixed Threshold" and "Dynamic Threshold" schemes proposed in [25]. In these benchmark schemes, the ONU continues the sleep mode until its queue length reaches a certain threshold. This threshold is fixed at 100 frames for the Fixed Threshold scheme while in the Dynamic Threshold scheme, the threshold is dynamically updated in each interval with increment step 1 frame and targeted delay 6 ms [25]. As explained in [26], our considered traffic model is more realistic than the one used in [25]. In each figure, the marked points show the average values for a high number of runs, each having a standard deviation less than 5%.

Fig. 3 shows the performance metrics versus the offered load Ω for different values of D_i, which have been commonly used in the literature for performance validation. As expected, the power efficiency decreases with the offered load Ω in Fig. 3(a), since when the system is highly loaded, there are more accumulated data packets in the ONU queues and we need more power to serve them. The power efficiency of our scheme depends on D_i and can be more or less than the benchmarks. Similarly, depending on the values of Ω and D_i, the proposed scheme provides lower or more transmission delay than the benchmark methods, as can be seen in Fig. 3(b). At low loads, the Dynamic Threshold scheme offers a better transmission delay at the cost of lower power efficiency. Conversely, at high loads, the proposed scheme delays packets less by consuming a bit more power than the Dynamic Threshold method. There is a mismatch of about 4 ms between the reported delays in Fig. 3(b) and the set values D_i, which is due to the extra delay imposed by the collecting and shaping buffers. Clearly, the proposed scheduling offers smoother average delay curves, especially compared to the Fixed Threshold scheme. This is a significant benefit of the proposed method, where the scheduling accepts the transmission delay as an input constant, and then guarantees it at every offered load Ω. In fact, our scheme seems more capable of supporting dynamic traffic conditions since it provides almost constant transmission delay at every load Ω. Consequently, the scheme can be used as a key enabler for low latency 5G transport over PONs, which requires constant and low transmission delays [5], [6], [28]. A significant feature is that the proposed scheme can trade power efficiency for transmission delay. In fact, when the ONUs can tolerate a higher transmission delay, the arrived packets can accumulate more in the delaying buffer to let the ONUs operate more in sleep mode and consume less power. When $\Omega = 0.5$, this tradeoff improves the power efficiency by more than 100% if the ONUs can tolerate a transmission delay of $D_i = 18$ ms rather than $D_i = 10$ ms. As can be seen in Fig. 3(c), there is no packet drop for $V_i = 100, \forall i$. This strict condition on zero drop rate creates a sudden increase for the transmission delay curves near $\Omega = 1$ in Fig. 3(b).

As illustrated in Fig. 4(c), if the ONUs can tolerate a certain amount of drop rate at high loads, we can decrease V_i to add a controllable amount of packet drops and improve the smoothness of the delay curves at high loads without changing the expected power efficiency, as depicted in Figs. 4(a) and 4(b). For more stringent D_i constraints, a higher drop rate should be tolerated at high loads, as can be seen in Fig. 4(c). Incorporating the drop rate in the scheduling offers beneficial tradeoffs and results beyond the capabilities of the benchmark methods.

Fig. 5 illustrates how diversity can improve the system performance, where for each curve, a certain portion of the ONUs have $D_i = 6$ ms while for the other ONUs $D_i = 14$ ms. Depending on the delay distribution among the ONUs, the power efficiency and delay curves fall between the corresponding curves for the two extreme cases with $D_i = 6$ ms and $D_i = 14$ ms for all ONUs. As the number of ONUs with $D_i = 14$ ms increases, the power efficiency improves since the ONUs with $D_i = 14$ ms can operate in sleep mode for longer durations. This improvement results from the inherent capability of the proposed scheme that can separately take into account the diverse transmission delays of the ONUs.

Infinite storage capacity is an impractical assumption of many dynamic bandwidth allocation methods such as the benchmarks. However, our proposed buffering structure assumes a finite storage capacity. Fig. 6 investigates the impact of the limited storage capacity of the delaying buffer Q_i on the performance metrics for a targeted transmission delay $D_i = 6$ ms, $\forall i$. The storage capacity Q_i has a negligible effect on the power efficiency and transmission delay. However, when the storage capacity is low, the strict transmission delay of $D_i = 6$ ms is achieved by dropping a number of packets at high offered loads Ω. As shown in Fig. 6(c), a storage capacity of $Q_i = 80$ Mbit, $\forall i$ can bring the drop-rate to zero.

Service prioritization is an important requirement, which is well supported by the proposed scheme. Fig. 7 shows the performance curves for a scenario including four distinct QoS classes with different offered loads and service priorities. Each class includes 8 ONUs. Class 1 and 3 with the lower penalty coefficient $V_i = 1$ have less service priority compared to the two other classes with $V_i = 100$. Class 3 and 4 have the same traffic load, twice the traffic loads of class 1 and 2. As shown in Fig. 7(a), class 1 and 2 with lower traffic load provide better

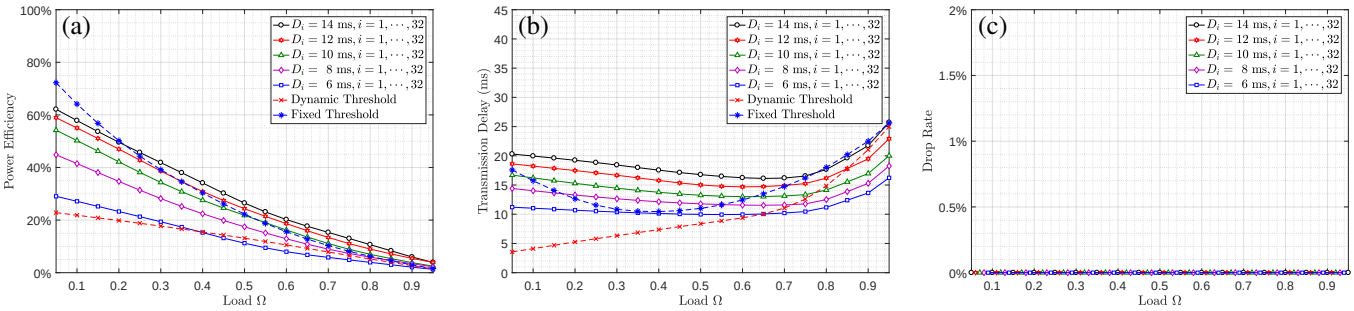

Fig. 3: Performance evaluation for various values of the transmission delay D_i and a high drop penalty $V_i = 100, \forall i$.

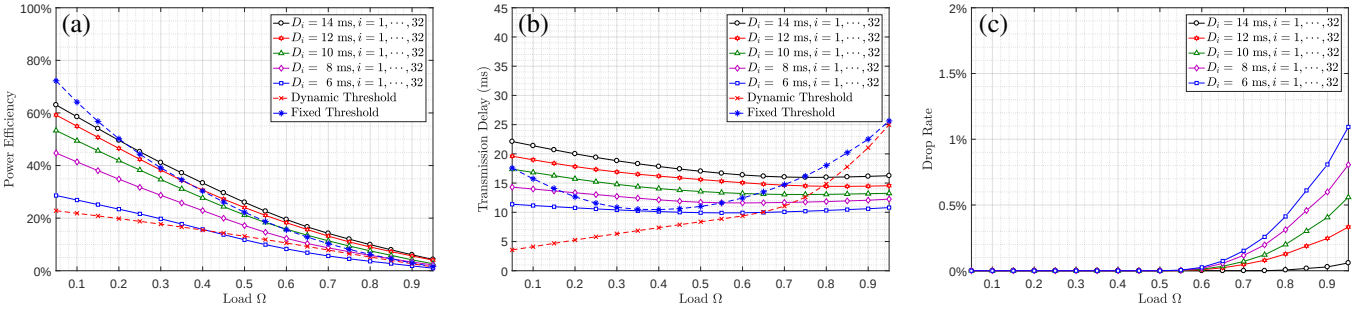

Fig. 4: Performance evaluation for various values of the transmission delay D_i and a low drop penalty $V_i = 1, \forall i$.

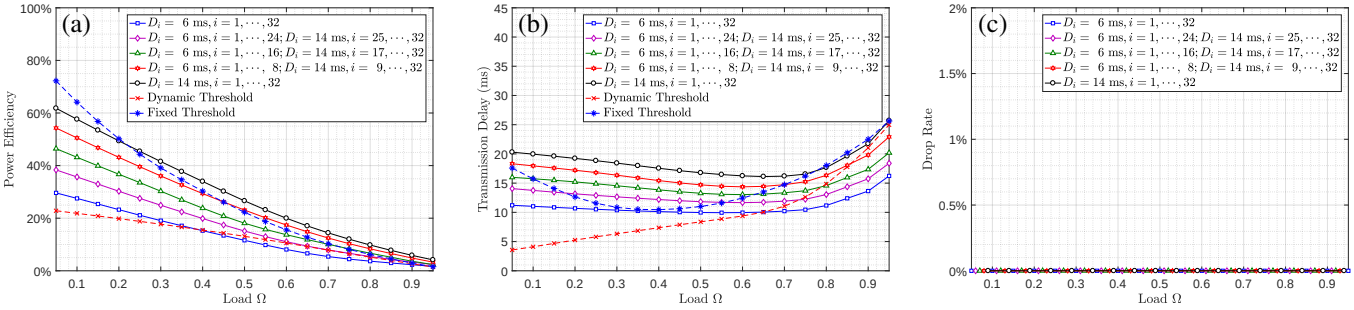

Fig. 5: Performance evaluation for a diverse scenario, where some of the ONUs have the transmission delay $D_i = 14$ ms and the others $D_i = 6$ ms.

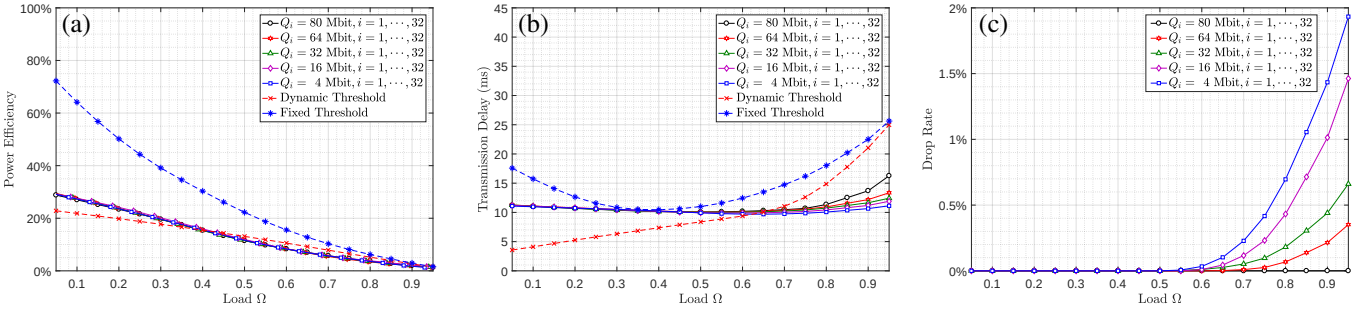

Fig. 6: Performance evaluation for various values of the delaying buffer size Q_i and fixed transmission delay $D_i = 6$ ms, $\forall i$.

power efficiency. Regardless of the class type, all the classes experience almost the same transmission delay, as can be seen in Fig. 7(b). However, at high loads, the performance of class 2 and 4 with higher priority is guaranteed at the cost of a little bit drop for classes 1 and 3 with lower drop penalty coefficient. Different offered load to classes 1 and 3 or 2 and 4 does not affect service prioritization.

The specified results and statements also hold for the proposed TWDM-PON scheduling scheme with $N_W \geqslant 1$ wavelengths. However, with N_W wavelengths, the PON almost sustains N_W times higher offered load Ω, as shown in Fig. 8, where load $\Omega = 1$ is the offered traffic $\sum_{i \in \mathbb{Z}_1^I} \overline{\omega_i}$ that fills the upload transmission capacity of a wavelength R_U. The same power efficiency, transmission delay, and drop rate

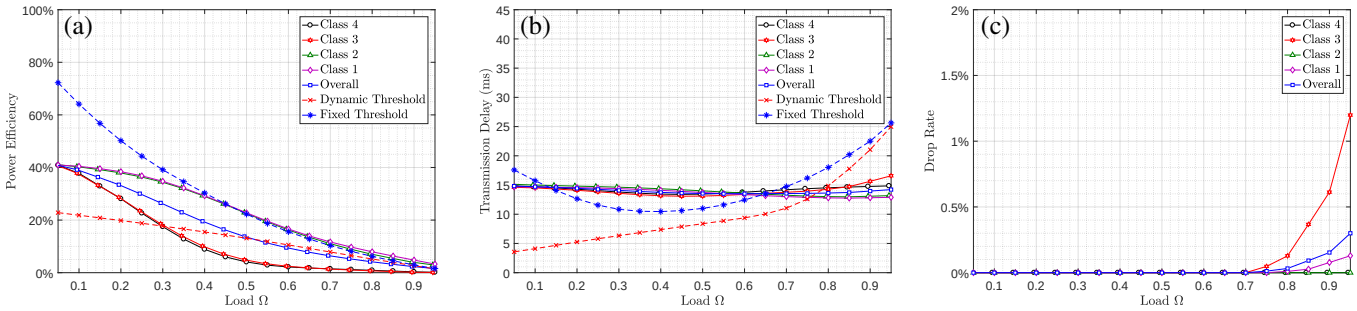

Fig. 7: Performance evaluation for four different QoS classes, where class 1, 2, 3, and 4 own ONUs $i \in \mathbb{Z}_1^8$, $i \in \mathbb{Z}_9^{16}$, $i \in \mathbb{Z}_{17}^{24}$, $i \in \mathbb{Z}_{25}^{32}$, respectively. Classes 1 and 3 have $V_i = 1, i \in \mathbb{Z}_1^8 \cup \mathbb{Z}_{17}^{24}$, class 2 and 4 have $V_i = 100, i \in \mathbb{Z}_9^{16} \cup \mathbb{Z}_{25}^{32}$, and class 3 and 4 have twice the traffic load of class 1 and 2 equal to $\overline{\omega_i}/R_U, i \in \mathbb{Z}_1^{16}$.

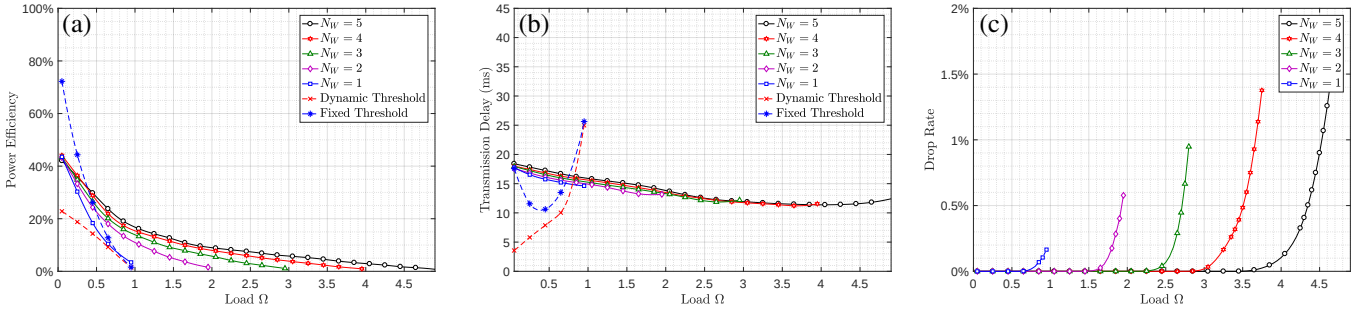

Fig. 8: Performance evaluation for various numbers of wavelength N_W and fixed transmission delay $D_i = 12$ ms, $\forall i$ and drop penalty $V_i = 1, \forall i$.

are roughly obtained for $\Omega \leqslant 1$ and every N_W. When $\Omega > 1$, all ONUs should be active almost always, possibly with different working wavelengths, to serve the incoming traffic load; therefore, the power efficiency is very low. A TWDM-PON with N_W wavelengths can roughly provide a constant transmission delay up to the load $\Omega = N_W$, where the drop rate or transmission delay increases suddenly, depending on the value of the drop penalty V_i, as depicted in Figs. 8(b) and 8(c).

V. Conclusions

We use stochastic optimization to propose a general scheduling framework for PONs, and employ Lyapunov optimization to establish a structured technique for solving the proposed formulation. The application of the proposed scheme for transmission scheduling in TDM- and TWDM-PONs is discussed and demonstrated. Particularly, we develop a sample power-efficient QoS-aware scheduling procedure for TDM- and TWDM-PONs. The example scheduling method is completely described by closed-form mathematical expressions and accepts the required transmission delays as its input parameters. According to the simulation results, the proposed scheme consumes the least power to approximately offer a constant transmission delay at every offered load. The level of smoothness in the transmission delay curves can be controlled by adjusting the buffer capacity or packet drop rate. Moreover, distinct QoS classes with different service priorities and features can be defined. A significant capability of the scheme is its inherent inclusion of service diversity, which enables a typical improvement of 50% in power efficiency at an offered load of 0.5, if half of the subscribers require a strict transmission delay of 10 ms while the other subscribers have a looser condition of 18 ms. Furthermore, using the embedded tradeoffs of the scheme, the consumed power can be reduced by 20% at an offered load of 0.5 if all subscribers can tolerate an average transmission delay of 18 ms rather than 10 ms. The applications of the proposed scheme for transmission scheduling in other possible types of NG-PONs, which might be proposed, demonstrated, or deployed in near future, can be a potential research topic.

References

[1] C. F. Lam, *Passive Optical Networks: Principles and Practice*. Elsevier, 2011.

[2] L. R. Dos Santos, F. R. Durand, A. Goedtel, and T. Abrão, "Auto-tuning PID distributed power control for next-generation passive optical networks," *Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D110–D125, 2018.

[3] D. M. Dourado, R. J. L. Ferreira, M. de Lacerda Rocha, and U. R. Duarte, "Energy consumption and bandwidth allocation in passive optical networks," *Optical Switching and Networking*, vol. 28, pp. 1–7, 2018.

[4] J. S. Wey, J. Zhang, X. Lu, Z. Ma, and B. Chen, "Real-time investigation of transmission latency of standard 4K and virtual-reality videos over a commercial PON testbed," in *Optical Fiber Communication Conference (OFC)*, 2018, pp. Tu2G–3.

[5] J. S. Wey and J. Zhang, "Passive optical networks for 5G transport: technology and standards," *Journal of Lightwave Technology*, vol. 37, no. 12, pp. 2830–2837, 2018.

[6] Y. Nakayama and D. Hisano, "Wavelength and bandwidth allocation for mobile fronthaul in TWDM-PON," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7642–7655, 2019.

[7] S. S. Lee, K. Li, and M. Wu, "Design and implementation of a GPON-based virtual OpenFlow-enabled SDN switch," *Journal of Lightwave Technology*, vol. 34, no. 10, pp. 2552–2561, 2016.

[8] D. Nesset, "PON roadmap," *Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. A71–A76, 2017.

[9] S. Dutta, D. Roy, C. Bhar, and G. Das, "Online scheduling protocol design for energy-efficient TWDM-OLT," *Journal of Optical Communications and Networking*, vol. 10, no. 3, pp. 260–271, 2018.

[10] M. P. I. Dias, B. S. Karunaratne, and E. Wong, "Bayesian estimation and prediction-based dynamic bandwidth allocation algorithm for sleep/doze-mode passive optical networks," *Journal of Lightwave Technology*, vol. 32, no. 14, pp. 2560–2568, 2014.

[11] E. Wong, M. P. I. Dias, and L. Ruan, "Predictive resource allocation for Tactile Internet capable passive optical LANs," *Journal of Lightwave Technology*, vol. 35, no. 13, pp. 2629–2641, 2017.

[12] M. Hadi and M. R. Pakravan, "Analysis and design of adaptive OCDMA passive optical networks," *Journal of Lightwave Technology*, vol. 35, no. 14, pp. 2853–2863, 2017.

[13] ——, "Rate-maximized scheduling in adaptive OCDMA systems using stochastic optimization," *IEEE Communications Letters*, vol. 22, no. 4, pp. 728–731, 2018.

[14] M. Hadi and E. Agrell, "Iterative configuration in elastic optical networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, 2020.

[15] M. J. Neely, "Stochastic optimization for Markov modulated networks with application to delay constrained wireless scheduling," in *Conference on Decision and Control (CDC)*, 2009, pp. 4826–4833.

[16] ——, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[17] R. A. Butt, S. M. Idrus, K. N. Qureshi, N. Zulkifli, and S. H. Mohammad, "Improved dynamic bandwidth allocation algorithm for XGPON," *Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 87–97, 2017.

[18] L. G. Zulai, F. R. Durand, and T. Abrão, "Energy-efficient next-generation passive optical networks based on sleep mode and heuristic optimization," *Fiber and Integrated Optics*, vol. 34, no. 3, pp. 91–111, 2015.

[19] X. Li, K. Kanonakis, N. Cvijetic, A. Tanaka, C. Qiao, and T. Wang, "Joint bandwidth provisioning and cache management for video distribution in software-defined passive optical networks," in *Optical Fiber Communication Conference (OFC)*, 2014, pp. Tu3F–5.

[20] A. Dixit, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Energy efficient dynamic bandwidth allocation for Ethernet passive optical networks: Overview, challenges, and solutions," *Optical Switching and Networking*, vol. 18, pp. 169–179, 2015.

[21] G. V. Arevalo, R. C. Hincapie, and R. Gaudino, "Optimization of multiple PON deployment costs and comparison between GPON, XGPON, NGPON2, and UDWDM PON," *Optical Switching and Networking*, vol. 25, pp. 80–90, 2017.

[22] L. Kleinrock, *Queueing Systems: Computer Applications*, 5th ed. New York: Wiley-Interscience, 1976, vol. II.

[23] M. P. I. Dias, D. P. Van, L. Valcarenghi, and E. Wong, "Energy-efficient framework for time and wavelength division multiplexed passive optical networks," *Journal of Optical Communications and Networking*, vol. 7, no. 6, pp. 496–504, 2015.

[24] S. Bindhaiq, A. S. M. Supa, N. Zulkifli, A. B. Mohammad, R. Q. Shaddad, M. A. Elmagzoub, and A. Faisal, "Recent development on time and wavelength-division multiplexed passive optical network for next-generation passive optical network stage 2 (NG-PON2)," *Optical Switching and Networking*, vol. 15, pp. 53–66, 2015.

[25] S. Herrería-Alonso, M. Rodríguez-Pérez, M. Fernández-Veiga, and C. López-García, "On the use of the doze mode to reduce power consumption in EPON systems," *Journal of Lightwave Technology*, vol. 32, no. 2, pp. 285–292, 2013.

[26] C. Bhar, N. Chatur, A. Mukhopadhyay, G. Das, and D. Datta, "Designing a green optical network unit using ARMA-based traffic prediction for quality of service-aware traffic," *Photonic Network Communications*, vol. 32, no. 3, pp. 407–421, 2016.

[27] A. R. Dhaini, C. M. Assi, M. Maier, and A. Shami, "Dynamic wavelength and bandwidth allocation in hybrid TDM/WDM EPON networks," *Journal of Lightwave Technology*, vol. 25, no. 1, pp. 277–286, 2007.

[28] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 477–490, 2020.

APPENDIX A
CLOSED-FORM SOLUTION OF (3)

Since $d_i = b_i = 0$ for $i \notin \mathbb{A}$, without loss of generality, it is assumed that $\mathbb{A} = \mathbb{Z}_1^I$. Substituting $x_i = V_i + p_i D_i/(T_C\Gamma) \geqslant 0$, $y_i = a_i + q_i - \min\{Q_i, D_i a_i/T_C\}$, and $z = R_U\big(T_C - T_D - |\mathbb{A}|\,(T_H + T_G)\big) \geqslant 0$ in (3), we get

$$\min_{b_i, d_i, i\in\mathbb{A}} \quad \sum_{i\in\mathbb{A}} (b_i + x_i d_i) \quad \text{s.t.} \tag{7a}$$

$$b_i \geqslant 0, d_i \geqslant 0, \quad i \in \mathbb{A} \tag{7b}$$

$$y_i - b_i - d_i \leqslant 0, \quad i \in \mathbb{A} \tag{7c}$$

$$\sum_{i\in\mathbb{A}} b_i - z \leqslant 0. \tag{7d}$$

Combining $d_i \geqslant 0$ with (7c) yields $d_i \geqslant \max\{0, y_i - b_i\}$. To satisfy this bound with equality for all $i \in \mathbb{A}$ is optimum. Hence, d_i can be removed from the optimization as follows.

$$\min_{b_i, i\in\mathbb{A}} \quad \sum_{i\in\mathbb{A}} (b_i + x_i \max\{0, y_i - b_i\}) \quad \text{s.t.} \tag{8a}$$

$$b_i \geqslant 0, \quad i \in \mathbb{A} \tag{8b}$$

$$\sum_{i\in\mathbb{A}} b_i \leqslant z. \tag{8c}$$

For all $i \in \mathbb{A}$ such that $x_i \leqslant 1$, $b_i = 0$ is clearly optimal. Similarly, for all $i \in \mathbb{A}$ such that $y_i \leqslant 0$, $b_i = 0$ is also optimal. When $y_i > 0$, then only $b_i \leqslant y_i$ needs to be considered, since any $b_i > y_i$ will give a higher objective value and reduce the slack in (8c). Hence, when $x_i > 1$ and $y_i > 0$, it is optimal to set $b_i = y_i$ for as many i values as (8c) permits, with priority given to those with highest x_i.

Formally, assume that $\mathcal{X}(h), h = 1, \ldots, |\mathbb{A}|$ returns the index of the variable having the hth largest coefficient x_i in the active set $\mathbb{A}$. We have

$$b_{\mathcal{X}(1)} = \mathcal{I}\{y_{\mathcal{X}(1)} > 0\}\mathcal{I}\{x_{\mathcal{X}(1)} > 1\} \min\{y_{\mathcal{X}(1)}, z\}, \tag{9a}$$

and sequentially, for $h = 2, \ldots, |\mathbb{A}|$,

$$b_{\mathcal{X}(h)} = \mathcal{I}\{y_{\mathcal{X}(h)} > 0\}\mathcal{I}\{x_{\mathcal{X}(h)} > 1\} \min\{y_{\mathcal{X}(h)}, z - \sum_{h'=1}^{h-1} b_{\mathcal{X}(h')}\}, \tag{9b}$$

where $\mathcal{I}\{.\}$ is the indicator function returning 1 for a true argument, and otherwise 0. Further,

$$d_i = \max\{0, y_i - b_i\}, i \in \mathbb{A}. \tag{9c}$$